IDA

INSTITUTE FOR DEFENSE ANALYSES

Schedule Assessment Methods for Ballistic Missile Defense Ground-Based Software Development

Bruce R. Harmon, Project Leader
Neang I. Om

August 2003
Approved for public release;
distribution unlimited.
IDA Paper P-3600
Log: H 01-000569

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
08-2003

2. REPORT TYPE
Final

3. DATES COVERED *(From - To)*
Jan 2000-Aug 2003

4. TITLE AND SUBTITLE
Schedule Assessment Methods for Ballistic Missile Defense Ground-Based Software Development

5a. CONTRACT NUMBER
DASW01 98 C 0067

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S), Last Names First
Harmon, Bruce R.
Om, Neang I.

5d. PROJECT NUMBER

5e. TASK NUMBER
DB-2-597.18

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Institute for Defense Analyses
4850 Mark Center Drive
Alexandria, VA 22311-1882

8. PERFORMING ORGANIZATION REPORT NUMBER
IDA Paper P-3600

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Missile Defense Agency
2120 Washington Blvd.
Suite 100
Arlington, VA 22204

10. SPONSOR/MONITOR'S ACRONYM(S)
MDA/PIV

11. SPONSORING/MONITORING AGENCY REPORT NUMBER

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This report documents software development schedule models for ground-based segments of ballistic missile defense. The objective was to provide methods for assessing the reasonableness of proposed acquisition schedules for ground-based software of proposed system architectures. IDA developed time estimating relationships (TERs) using historical data from the Air Force Space and Missile Command (SMC) database. TERs are estimated using least-square regression analysis where the relevant duration in months is the dependent variable. Schedule drivers are the independent variables. Schedule drivers used include the rate at which resources are applied to the project specified as average staffing level as well as software size in source lines of code. This approach will help answer the question of how much program duration can be shortened with added staff while holding project size constant.

15. SUBJECT TERMS
Time Estimating Relationships; Computer Programs; Ballistic Missile Defense

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
Unclassified	Unclassified	Unclassified	Same as Report	30

19a. NAME OF RESPONSIBLE PERSON (Last Name First)
Vickers, Scott M.

19b. TELEPONE NUMBER (*Include area code*)
(703) 553-5697

This work was conducted under contract DASW01 98 C 0067, Task DB-2-597, for the Missile Defense Agency. The publication of this IDA document does not indicate endorsement by the Department of Defense, nor should the contents be construed as reflecting the official position of that Agency.

INSTITUTE FOR DEFENSE ANALYSES

IDA Paper P-3600

Schedule Assessment Methods for Ballistic Missile Defense Ground-Based Software Development

Bruce R. Harmon, Project Leader
Neang I. Om

Preface

The Institute for Defense Analyses (IDA) prepared this paper for the Ballistic Missile Defense Organization (since redesignated the Missile Defense Agency) under a task entitled “Methods To Assess Schedules for the Strategic Defense System.” The objective of the task was to develop analytical tools for assessing proposed schedules for ground-based ballistic missile defense elements. This paper fulfills that objective by providing time-estimating relationships for assessing the reasonableness of such schedules.

John W. Bailey and Reginald N. Meeson of IDA were the technical reviewers for this paper.

Table of Contents

List of Figures

List of Tables

I. Introduction

A. Background

Software development is an increasingly important part of weapon systems acquisition for the Department of Defense (DoD). In previous studies for the Ballistic Missile Defense Organization—since redesignated the Missile Defense Agency (MDA)—the Institute for Defense Analyses (IDA) focused on software development cost and schedule models for space-based systems [1]. In this study, we analyze software development schedules relevant to ground-based ballistic missile command, control, and communications (C^3). This study is part of an IDA task to develop models to assess acquisition schedules for MDA programs. Because ground-based battle management C^3 is a critical path item in the development of MDA architectures, its software development schedule is also important.

B. Approach

The focus of the study was to analyze existing databases that describe a large sample of historical software development efforts. For that purpose, we used the Space and Missile Systems Center (SMC) software database. The SMC database contains software development information of past programs submitted by contractors and data collected by SMC.[1]

We examined data for the DoD programs in the SMC database and developed time-estimating relationships (TERs) to estimate software development schedules. TERs were estimated using least-squares regression analysis, where the dependent variable was the schedule interval in months.

The traditional approach to estimating software development schedule is to derive an equation that uses software size in source lines of code (SLOC) as the single independent variable. In addition to the software size, our analysis used the rate at which resources are applied to the project (specified as average staffing level) and the type of software application to assess the development

---

1 The SMC database was known previously as the Space Systems Cost Analysis Group (SSCAG) database, which was maintained by Management Consulting & Research, Inc.

schedule. Our approach will help determine how much a program's duration can be shortened with added staff while holding project size (SLOC) constant.

C. Method

To develop the TERs for estimating software development schedules, we used the same method used in previous IDA work for MDA [1]. Traditionally, software development TERs are based on the assumption that schedules are related to software size in exponential form [2]:

$$Duration = A \times (Size)^B \tag{1}$$

This specification takes into account non-linearity with respect to the independent variable, while its estimation can be performed using linear regression with log transformation.

In this equation, development time (*Duration*) is measured in number of months required to develop the software. The coefficient *A* is the intercept term derived through a log-transformed regression. The input *Size* is measured by the number of SLOC. The exponent *B* is derived from the regression analysis. In addition to the traditional size schedule driver, we examined average staffing level and type of software application (C^3, mission planning, signal processing, test and simulation, etc.).

D. Report Organization

This report is divided into five short chapters and an appendix. Following this introduction (Chapter I), Chapter II describes data evaluation and normalization and the method used to analyze the software schedule. Chapter III documents the results of our TER development, Chapter IV discusses application of the models, and Chapter V summarizes the findings. The appendix discusses the possible effect on schedules of radar transmit/receive (T/R) module availability.

II. Data

A. Database

The Space and Missile Systems Center (SMC) database contains data describing ground segment and embedded flight software used in various aerospace and defense systems, including the space shuttle. The database has data from 22 member companies, including SMC. It includes actual and estimated values of software development cost, size, and schedule. The SMC database contains software development programs at the Computer Software Configuration Item (CSCI) and project levels. A project is often made up of multiple CSCIs. The sizes were measured in source lines of code (SLOC) and effort, in staff-months. The durations, measured in months, were calculated from the schedule. In this study, we focus on ground segment software only.

B. Data Evaluation and Normalization

In constructing the database for analysis, we were looking for records that had the required data for ground-based software. We checked each record for correct basing mode (where the software resides—ground, ground in support of space, space, or air), software type (C^3, signal processing, or other), size in SLOC, and effort in staff-months. Since we were seeking to understand TERs for ballistic missile C^3 software development, we were interested in only the ground-related programs. We used only those data points that had actual values for schedule, size, and effort; data records with estimated values were excluded.

The SMC database contains over 3,000 records, 1,332 of which have ground-related data. Only 660 of the ground-related records have both size and effort in staff-months. The final ground segment database for the study included only ground and military mobile (ship and ground vehicle) programs.

Our data normalization process had three steps, as follows:

1. We included only data points that had size and effort.
2. We excluded data points that were not at the CSCI or project level.
3. We eliminated data points that did not have dates for System Requirements Review (SRR) and either First Qualification Test (FQT) or

Operational Test and Evaluation (OTE). Figure 1 maps these milestones to the development process.

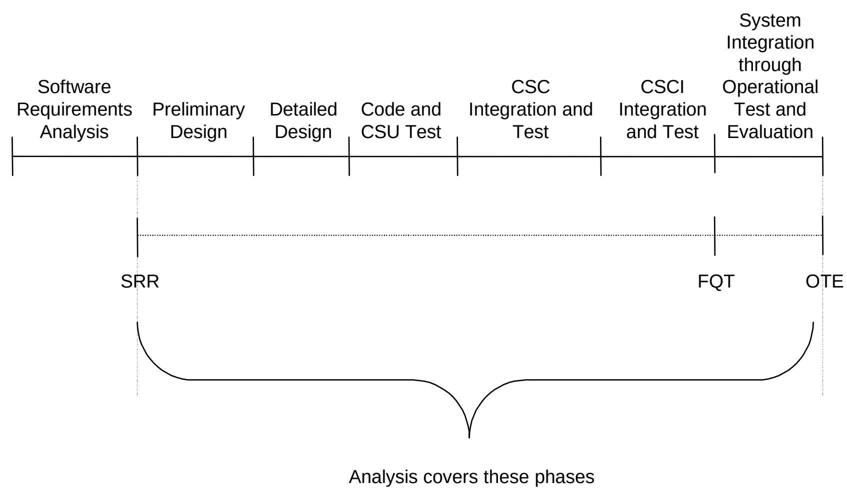


Figure 1. Software Development Phases

After this elimination process, we had 79 data points at the CSCI level and 19 at the project level. For our analysis, we classified these data into the following two categories:

- *C^3 and related software*—mission control, command processing, network monitoring, network control and switching, sensor control, message/signal processing, process control, and diagnostics.
- *Other software*—mission planning, database, test and simulation, man-machine interface (MMI) graphics, and office automation.

We began by analyzing the data at the CSCI level. The regression analysis shows an unacceptable goodness of fit even though a few outliers were eliminated and the independent variables were applied in different ways. We then analyzed the 19 observations at the project level (using the data in Table 1, presented later in this chapter). This analysis produced an acceptable statistical significance and model fit for the variables used.

We suspect that for the CSCI data, the data-entry personnel used the wrong schedule phase definitions for most schedule milestones. For example, the CSCI database shows many CSCIs within the same project with the same schedule milestones despite big differences in size. We therefore developed the TERs for ground-based software using only the observations at the project level.

C. Equivalent Source Lines of Code

Software size is the primary driver for software schedule; therefore, a convention to normalize the size of a CSCI, including reused and modified software, is important.

We measured software size in SLOC as presented in the SMC database. A SLOC is defined as a single instruction, not necessarily a physical line [3]. Included are data declaration statements, mathematical statements ($i = j + 1$), conditional statements (if, then, else), job-control language, data typing statements, and input/output format statements. Excluded are comment statements, blank lines, non-delivered programmer debug statements, continuation of format statements, machine- or library-generated data statements, commercial off-the-shelf software, and in-house software. We adjusted the software size for whether the code was reused or modified using the method documented in [4]:

$$ESLOC = New\ SLOC + 0.5\ Modified\ SLOC + 0.25\ Inherited\ SLOC,$$

where

ESLOC = equivalent source lines of code;

New SLOC = newly developed source lines of code;

Inherited SLOC = source lines of code reused without changes; and

Modified SLOC = reused source lines of code changed by the project.

D. Average Staff Size

To obtain average staff size, we divided total effort in staff-months by the development duration in months. We assumed total effort in staff-months and the duration from SRR to FQT to be as indicated in the SMC database. All effort data in the database is for SRR to FQT only, even when no FQT date is provided. Therefore, we normalized the average staff calculation to use SRR to FQT effort and SRR to FQT duration. Figure 2 illustrates the effort phase and development schedules. For those data records with only the OTE duration, we used an adjustment procedure explained in Chapter III, subsection C.3.

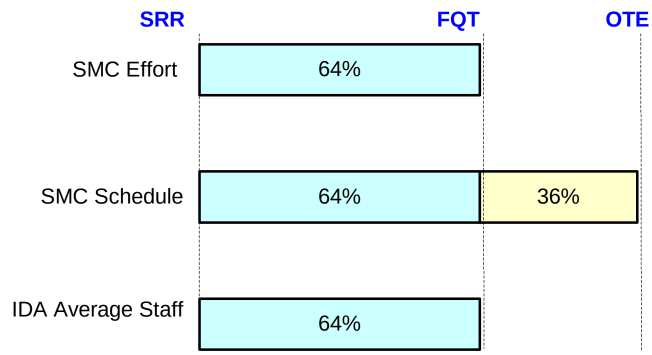


Figure 2. Average Staff Level versus SMC Effort and Schedule Phases

Table 1 presents the observations at the project level used for this analysis, including ESLOC and average staff size.

Table 1. Ground-Based Software at the Project Level

Project	Total Effort (staff-months)	ESLOC	Average Staff Size	Duration (months) SRR to FQT	SRR to OTE	Operating Environment	Application
1[a]	19,079	709,000	577	33	57	Military ground	Command and control
2	3,285	799,406	68	—	75	Military mobile (van/ship)	Command and control
3	2,131	306,773	222	—	15	Military ground	Test
4	1,613	622,129	70	23	—	Military ground	Mission planning
5[a]	709	160,000	39	18	26	Military ground	Office automation
6	621	144,463	18	34	—	Military ground	Process control
7[a]	658	70,143	21	31	58	Military ground	Command and control
8	230	48,000	14	17	—	Military mobile (van/ship)	Process control
9	194	135,362	30	—	10	Military ground	Software development tools
10	181	141,350	4	50	—	Military mobile (van/ship)	Process control
11	170	13,000	12	—	22	Military ground	Command and control
12	168	36,362	8	—	32	Military ground	Database
13[a]	103	20,000	10	10	14	Military ground	Test
14	36	8,000	1	—	27	Missile	Other
15	9	4,100	1	—	12	Military ground	Simulation

[a] These projects had data for both OTE and FQT duration.

III. Development of Time-Estimating Relationships

A. Approach

Because of the reasons outlined in the previous chapter, our approach was to analyze the data at the project level. Because there are not many good schedule data points at the project level, and we wanted to make use of as much data as possible, we combined data with FQT and OTE milestones. We had a total of 19 data points from 15 data records—7 had OTE data only, 4 had FQT data only, and 4 had both OTE and FQT data. Figure 3 depicts the relationship of these data records.

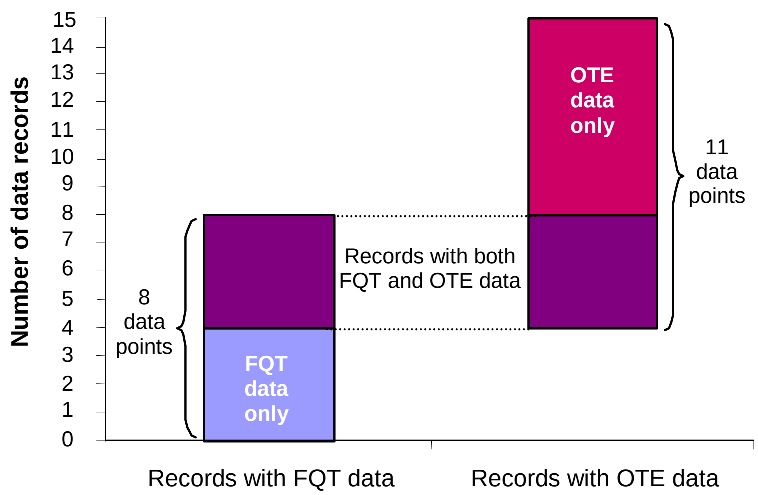


Figure 3. Relationship of Data Records

We pooled FQT and OTE using a dummy variable to distinguish between FQT and OTE durations as expressed in months from SRR. We also developed FQT and OTE models separately. Models generated by all three schemes are presented in this report.

B. Analysis Strategy

For this analysis, we used ordinary least squares (OLS) regression. Our TERs take on the form shown previously in equation (1). To estimate the coefficient and exponent, we transformed the equation to a logarithmic form and then applied OLS regression. Then we exponentiated *A* to transform the equation back from logarithmic form. When the equation is transformed from the logarithmic form back to its original form, the multiplicative residuals are assumed to be distributed log normally. Because the log normal distribution is right-skewed, the expected value and most likely value (mode) of the residuals are no longer equal. So an adjustment must be made for the multiplicative form to yield the expected value for the dependent variable.

We made this adjustment by adding one-half of the regression mean square error to the constant term of the logarithmic equation before it was transformed into the multiplicative form [5]. Then we transformed the intercept term into a multiplicative constant, which yields an adjustment factor (adjusted constant term/unadjusted constant term) on the multiplicative form greater than one. In presenting our TERs, we report the adjusted multiplicative equation along with the adjustment factor so that the equation can be adjusted back to yield the most-likely value.

C. Results

We tested several different specifications in developing the TERs. Three separate models were developed:

- Duration from SRR to OTE only.
- Duration from SRR to FQT only.
- Pooled SSR to FQT and OTE data using a dummy variable (1/0) to distinguish FQT durations from the OTE durations.

The dependent variable is the duration in months from SRR to completion (FQT or OTE). Independent variables are:

- Equivalent source lines of code (*ESLOC*),
- Average staff size (*AvgStaff*), and
- A dummy variable to distinguish software type (C^3). C^3 takes the value of 1 for C^3 software and 0 for all other software types.

1. Duration from SRR to FQT

Equation (2) presents the TER we developed using data for size, effort, and schedule dates from SRR to FQT. We had only 8 observations for this analysis.

$$\mathit{Duration\ SRR\ to\ FQT} = 0.306 \times \underset{(4.24,\ 0.013)}{\mathit{ESLOC}^{\,0.397}} \times \underset{(-2.67,\ 0.055)}{\mathit{AvgStaff}^{\,-0.200}} \times \underset{(3.87,\ 0.017)}{1.812^{\,C^3}} \qquad (2)$$

$N = 8$ Adjusted $R^2 = 0.83$ $SEE = 0.20$ Intercept Adjustment = 1.02

The *t* scores and probability levels are in parentheses below the parameter estimates. *N* is the number of observations. Adjusted R^2 is a coefficient of determination measuring the proportion of variation in the data explained by the model, adjusted for the number of independent variables in the regression. SEE is the standard error of the estimate. The intercept adjustment factor adjusts the intercept to yield the expected value for the dependent variable when the equation is transformed from the log-log form back to a multiplicative form (see Section B for details).

This model shows good levels of statistical significance for *ESLOC* (0.013) and C^3 (0.017), and an acceptable significance for *AvgStaff* (0.055). The model indicates that C^3 software development duration takes about 80% longer than other software types when counting from SRR to FQT phase. Figure 4 shows the relationship between actual and predicted development durations in months from SRR to FQT. The 45-degree line is the demarcation between data points that are underestimated and overestimated by the model.

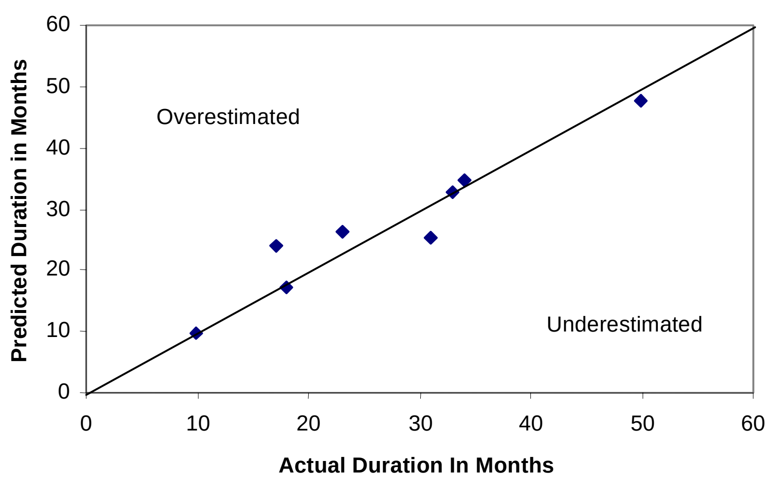


Figure 4. Actual versus Predicted Months from SRR to FQT

Table 2 presents the data used for the development of equation (2), along with the model-estimated values.

Table 2. Development Duration from SRR to FQT

Project	SRR to FQT (months)		ESLOC	Average Staff Size	C^3 Dummy
	Actual	Estimated			
1	33	33	709,000	577	1
4	23	26	622,129	70	70
5	18	17	160,000	39	0
6	34	35	144,463	18	1
7	31	25	70,143	21	1
8	17	24	48,000	14	1
10	50	48	141,350	4	1
13	10	10	20,000	10	0

2. Duration from SRR to OTE

Equation (3) presents the TER we developed for SRR to OTE. For this analysis, we used only data points that have software size and schedule dates from SRR to OTE. We did not make any adjustment to obtain more data points. For this model, we had 11 observations for the analysis.

$$\textit{Duration SRR to OTE} = 0.746 \times \underset{(1.87,\ 0.103)}{\textit{ESLOC}^{\,0.386}} \times \underset{(-1.55,\ 0.164)}{\textit{AvgStaff}^{\,-0.316}} \times \underset{(3.05,\ 0.018)}{2.52^{\,C^3}} \tag{3}$$

$N = 11$ Adjusted $R^2 = 0.57$ $SEE = 0.44$ Intercept Adjustment = 1.104

In this analysis, statistics show the worst fit of any of our models; neither *ESLOC* nor *AvgStaff* is significant at the 0.10 level. The type of software application proved to be more significant than the other independent variables with the coefficient of the software type variable suggesting that C^3 software durations are 2.5 times longer than other application types. Figure 5 shows the relationship between the actual and predicted software development durations for this TER.

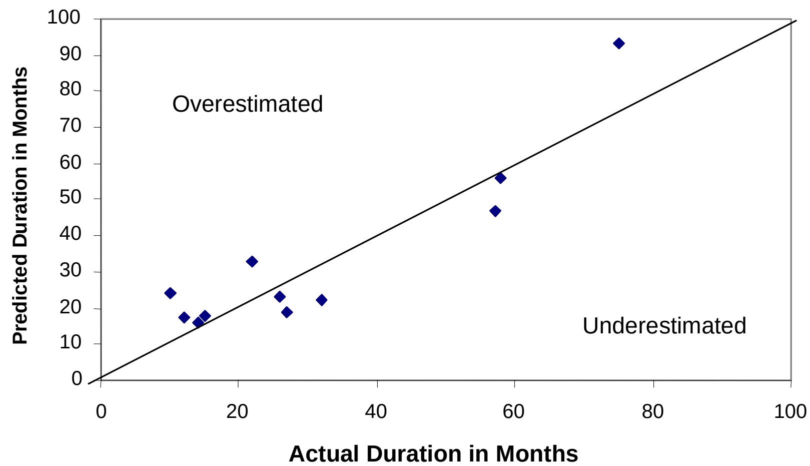


Figure 5. Actual versus Predicted Months from SRR to OTE

Table 3 presents the data used and the resulting estimated values from equation (3).

Table 3. Development Duration from SRR to OTE

Project	SRR to OTE (months) Actual	SRR to OTE (months) Estimated	ESLOC	Average Staff Size	C^3 Dummy
1	57	47	709,000	577	1
2	75	93	799,406	68	1
3	15	18	306,773	222	0
5	26	23	160,000	39	0
7	58	56	70,143	21	1
9	10	24	135,362	30	0
11	22	33	13,000	12	1
12	32	22	36,362	8	0
13	14	16	20,000	10	0
14	27	19	8,000	2	0
15	12	18	4,100	1	0

3. Duration from SRR to OTE and FQT

Here we pooled OTE-only and FQT-only databases along with the four records with both FQT and OTE dates. The database consists of 19 observations (4 records contain both FQT and OTE durations). The base for the TER is the duration from SRR to OTE; we also added a 1/0 dummy variable called *FQT* to indicate the duration from SRR to FQT.

Note that we calculated the average staff size based on effort from SRR to FQT in accordance with definitions in the SMC database. Although there are some data points with OTE dates, the database reports effort only from SRR to FQT. Since we pooled FQT-only and OTE-only data together, the average staff level for the entire database for this analysis should also be based on the duration from SRR to FQT. Therefore, for the OTE-only data points, we adjusted the duration used to estimate average staff size by 64% [the *FQT* coefficient in equation (4)] by estimating the equation iteratively. Besides the dummy variable for FQT, the independent variables in the equation are software size (*ESLOC*), average staff size in staff-months (*AvgStaff*), and a 1/0 dummy variable distinguishing software type (C^3).

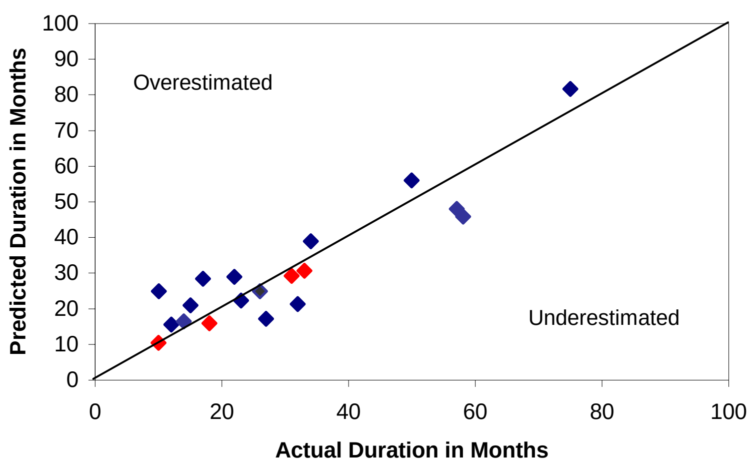
$$\mathit{Duration\ SRR\ to\ FQT/OTE} = 0.895 \times \mathit{ESLOC}^{\,0348} \times \mathit{AvgStaff}^{\,-\,0.230} \times 2.131^{\,C^3} \times 0.638^{\,FQT} \quad (4)$$

(3.16, 0.007) (–2.27, 0.039) (4.27, 0.000) (–2.49, 0.026)

$N = 19$ Adjusted $R^2 = 0.64$ $SEE = 0.36$ Intercept Adjustment = 1.067

All parameter estimates are significant at the 0.05 level.

Figure 6 shows the relationship between actual and predicted durations, and Table 4 presents the data used for the development of this relationship.



Note: Red denotes the data points with both FQT and OTE schedule data.

Figure 6. Actual versus Predicted Months from SRR to OTE and FQT

Equations (2) through (4) indicate that software development duration does not decrease in proportion to increases in staff level as denoted by the absolute exponent values of less than one (0.316, 0.200, and 0.230, respectively). This can be explained by the inefficiencies of a larger staff size discussed in [6].

Table 4. Development Duration from SRR to FQT and OTE

Project	SRR to FQT (months)		SRR to OTE (months)		ESLOC	Average Staff Size
	Actual	Estimated	Actual	Estimated		
1[a]	33	31	57	48	709,000	577
2	—	—	75	82	799,406	68
3	—	—	15	21	306,773	222
4	23	22	—	—	622,129	70
5[a]	18	16	26	25	160,000	39
6	34	39	—	—	144,463	18
7[a]	31	29	58	46	70,143	21
8	17	28	—	—	48,000	14
9	—	—	10	25	135,362	30
10	50	56	—	—	141,350	4
11	—	—	22	29	13,000	12
12	—	—	32	21	36,362	8
13[a]	10	10	14	16	20,000	10
14	—	—	27	17	8,000	1
15	—	—	12	16	4,100	1

[a] These projects had data for both OTE and FQT duration.

Counting from SRR to OTE, the coefficients of 2.52 and 2.131 on the C^3 variables of equations (3) and (4) suggest that C^3 software takes about 2 to 2.5 times longer than other ground software to develop when holding the software size and staff level constant. And, if counting from SRR to FQT, it takes about 1.8 times longer to develop C^3 software, as indicated by the C^3 coefficient of 1.812 in equation (2). The longer durations are probably due to a high degree of real-time processing and extremely high reliability of the C^3 programs when compared with other types in our database.

The 0.638 coefficient on the FQT dummy variable of equation (4) indicates the development duration from SRR to FQT takes, on average, about 64% of the development time from SRR to OTE. Also as expected, all three equations indicate that ground software development duration increases at about the same rates across the various models as software size increases [exponents of the *ESLOC* variable of 0.397 in equation (2), 0.386 in equation (3), and 0.348 in equation (4)].

D. Comparison with Boehm

To capture the plans and requirements phase not included in our TERs, we used Boehm's schedule distribution factors (Reference [2], p. 90) as a point of comparison. Table 5 presents Boehm's phase distribution factors for

development schedule in percentages for all basing modes. These factors vary with the project size in KSLOC.

Table 5. Schedule Phase Distribution: All Modes

Mode	Phase	Percentage of Schedule				
		Small (2 KSLOC)	Inter-mediate (8 KSLOC)	Medium (32 KSLOC)	Large (128 KSLOC)	Very Large (512 KSLOC)
Organic	Plans and requirements	10	11	12	13	—
	Product design	19	19	19	19	—
	Programming	63	59	55	51	—
	Integration and test	18	22	26	30	—
Semi-detached	Plans and requirements	16	18	20	22	24
	Product design	24	25	26	27	28
	Programming	56	52	48	44	40
	Integration and test	20	23	26	29	32
Embedded	Plans and requirements	24	28	32	36	40
	Product design	30	32	34	36	38
	Programming	48	44	40	36	32
	Integration and test	22	24	26	28	30

Source: Reference [2], p. 90.

In the following comparison, we chose a very large and complicated ground software project with 512 KSLOC or more, so we used the semi-detached distribution factors for a very large project to calculate the total development duration. Table 6 shows the data.

Boehm's model predicts the system integration phase will take 32% of the time from SRR to OTE. That is close to our FQT/OTE factor of 36% implied by the estimate of 0.64 on the FQT dummy variable. The TERs do not account for the plans and requirements phase. That phase is counted as time in addition to the development schedule, which usually consists of product design, programming, and system integration. For our example, an additional 24% is required for this early phase, according to Boehm.

Figure 7 compares the schedule phase distributions of the traditional Boehm models and our models for the semi-detached mode.

According to IDA's model, it takes about 36% of the total development time to develop ground software from FQT to OTE; with Boehm's model, it takes 32%. The difference might be because our data are for only military software programs, while Boehm's data include a variety of customers. Also, our models estimate software schedule at the project level. To estimate development duration using data at the CSCI level, the ESLOC must be integrated to the

project level before applying the models. However, if a CSCI is project-level integrated, there would be no need to integrate the ESLOC to apply the models.

Table 6. Schedule Phase Distribution: Semi-detached Mode (Ground-Based, >512 KSLOC)

Phase	Percentage of Schedule (SRR to OTE)
Plans and Requirements	24[a]
SRR to FQT	*68*[b]
Product Design	28
Programming	40
FQT to OTE	*32*
Integration and Test	32
Total (SRR to OTE)	*100*[c]

Source: Reference [2], p. 90.
[a] Percentage of schedule *not* accounted for by the TERs (i.e., percentage of SSR to FQT added).
[b] Percentage of schedule accounted for by SRR to FQT TER.
[c] Percentage of schedule accounted for by SRR to OTE TER.

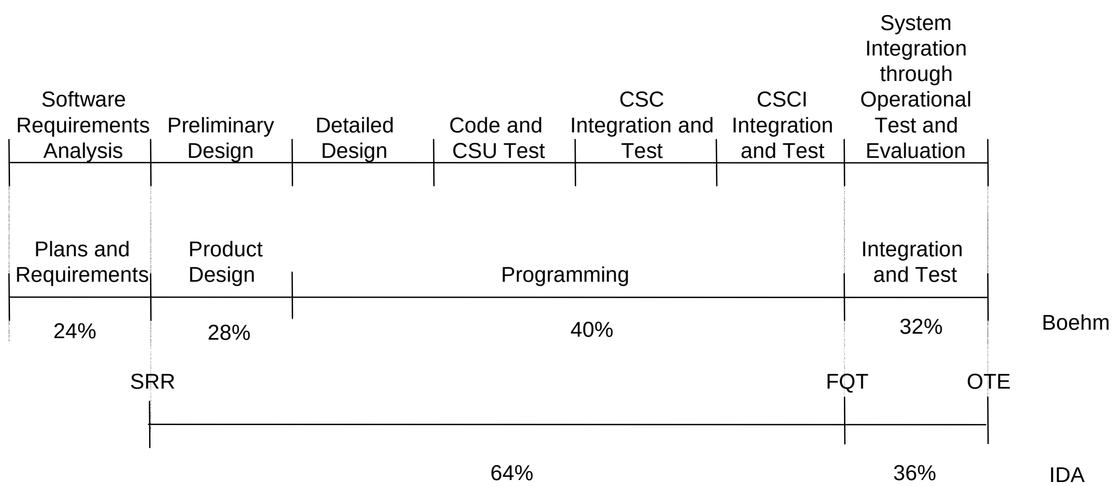


Figure 7. Schedule Phase Distribution (Semi-detached Mode, >512 KSLOC)

IV. Model Application

The time-estimating relationships in this paper can be applied in two ways. One approach is to use them in their role as a schedule assessment tool. Another approach is to estimate an independent variable in the model, given a desired schedule. In this chapter, we use a hypothetical ground-based program, the GBD-1, to illustrate both approaches. The hypothetical GBD-1 is a very large C^3 ground-based system being developed for the DoD. It requires 700,000 equivalent source lines of code (ESLOC) at the project level to carry out its mission.

Here we illustrate the application of the TERs in their roles as schedule assessment tools using equations (2), (3), and (4) from the previous chapter (repeated below).

$$\begin{aligned} \textit{Duration SRR to FQT} &= 0.306 \times \textit{ESLOC}^{\,0.397} \times \textit{AvgStaff}^{\,-0.200} \times 1.812^{\,C^3} \\ &= 0.306 \times 700{,}000^{\,0.397} \times 350^{\,-0.200} \times 1.812 \\ &= 37 \text{ months} \end{aligned} \tag{2}$$

$$\begin{aligned} \textit{Duration SRR to OTE} &= 0.746 \times \textit{ESLOC}^{\,0.386} \times \textit{AvgStaff}^{\,-0.316} \times 2.52^{\,C^3} \\ &= 0.746 \times 700{,}000^{\,0.386} \times 350^{\,-0.316} \times 2.52 \\ &= 53 \text{ months} \end{aligned} \tag{3}$$

$$\textit{Duration SRR to FQT/OTE} = 0.895 \times \textit{ESLOC}^{\cdot\,0348} \times \textit{AvgStaff}^{\,-0.230} \times 2.131^{\,C^3} \times 0.638^{\,FQT} \tag{4}$$

$$\begin{aligned} \textit{Duration SRR to FQT} &= 0.895 \times 700{,}000^{\,0.348} \times 350^{\,-0.230} \times 2.131^{1} \times 0.638^{\,1} \\ &= 34 \text{ months} \end{aligned}$$

$$\begin{aligned} \textit{Duration SRR to OTE} &= .895 \times 700{,}000^{\,0.348} \times 350^{\,-0.230} \times 2.131^{\,1} \times 0.638^{\,0} \\ &= 54 \text{ months} \end{aligned}$$

Table 7 presents schedule estimates derived using both IDA and Boehm FQT/OTE factors and assuming an average available staff of 350. The estimates include the additional 24% for the plans and requirements phase not accounted for by our models. For the FQT-only TER, the estimate includes an additional phase, system integration and test (36%).

Table 7. Schedule Estimation for the Hypothetical GBD-1 System (Semi-detached Mode, >512 KSLOC)

	Schedule (in months) by Phase				
	SRR to FQT	System Integration	SRR to OTE	Plans and Requirements	Total
Estimate using IDA Factors	(64%)	(36%)	(100%)	(24%)	
Equation (2)	37	21	58	14	72
Equation (3)	34	19	53	13	66
Equation (4)	34	20	54	13	67
Estimate using Boehm Factors	(68%)	(32%)	(100%)	(24%)	
Equation (2)	37	17	54	13	67
Equation (3)	36	17	53	13	66

Note: Red indicates months estimated by the models.

In estimating the staff level required given a fixed development duration, we also assumed data at the project level. For the GBD-1 program, the 700-KSLOC, ground-based software for C^3 must be completed in 68 months from SRR to OTE. What average staff level is needed to support that software development schedule?

From equation (3), to reach the OTE phase from SRR in 68 months instead of the 53 months in the above calculations, we computed the required average staff level as follows:

$$\textit{Duration SRR to OTE} = .746 \times \textit{ESLOC}^{\,0.386} \times \textit{AvgStaff}^{\,-0.316} \times 2.52^{\,C^3}$$

$$\textit{AvgStaff} = [.746 \times \textit{ESLOC}^{\,0.386} \times 2.52^{\,C^3} / \textit{Duration}]^{\,1/-0.316}$$

$$= [.746 \times (700)^{\,0.386} \times 2.52^{\,1} / 68]^{-1/0.316}$$

$$= 162$$

This suggests that a reduction by more than 50% in staff is possible by extending the schedule by only 25%. This illustrates the non-linearity between schedule compression and total effort.

This example shows that analysts can use the ground-based schedule assessment models presented in this report in different ways.

V. Observations

Analysis of the latest SMC software database yields very different results than from previous IDA studies of software schedules. In contrast to past studies, software schedules are less sensitive to software size or staff level. Software application type is an important schedule driver for ground-based software development. Given the same staff level and size, C^3 software takes about 2 to 2.5 times longer to develop than other types of ground software.

We hypothesized the following possible explanations for the results obtained:

- For historical programs, as characterized in the SMC database, software developments may not have been the critical path items.
- Software schedules may be more politically driven in military projects.
- Software development schedule reporting may be distorted by different assumptions about the definitions of software schedule phases.

Testing these hypotheses would require further investigation.

Appendix

Another area that could affect the schedules of MDA programs is the availability of radar transmit/receive (T/R) modules required for many surface-based elements of proposed system architectures. Possible bottlenecks and schedule delays could occur if T/R module production capacity is not sufficient to meet projected demand. This appendix presents a survey of T/R module producers and potential sources of demand.

Northrop Grumman and Raytheon both indicate that they have expanded their capacity to produce T/R modules in response to expectations of growth in certain programs, such as the Joint Strike Fighter (JSF), the F-22 fighter, and MDA programs. However, to this point they have produced relatively small quantities. Therefore, production capacity appears to be expanding faster than production.

Northrop Grumman's Electronic Sensors and Systems Sector (ESSS), a 1.6-million square foot facility near Baltimore, Maryland, includes the Advanced Microwave Electronics Center (AMEC), a 20,000 square foot facility. The AMEC includes three Clean Rooms and is certified for classified and Special Access Required (SAR) work. The manufacture of T/R modules accounts for 30% of AMEC's production. The AMEC facility has produced modules for a variety of programs, including ATF Demonstration/Validation; T/R ManTech; F-22 Engineering, Manufacturing, and Development (EMD); and Multifunction Integrated RF System/Multifunction Nose Array (MIRFS/MFA). The total production for these programs totals over 15,000 T/R modules

AMEC has expanded its capacity to include the JSF/MFA build (1,600 Twin Paks) and states that it can produce over 4,000 F-22 T/R modules per month on a two-shift basis. Including F-22 and space applications, AMEC expects to produce over one million Twin Paks (containing two T/R module assemblies) within 10 years.

Raytheon's T/R module business is based on, among other programs, JSF, Theater High Altitude Area Defense (THAAD), High Powered Discriminator-X (HPD-X), and F-18 radar (AN/APG-79). As in the case of Northrop Grumman, Raytheon indicates that it has expanded to be able to produce substantial quantities for these programs (especially MDA buys), but that none of the buys has yet materialized to a great extent. In fact, the MDA recently put the HPD-X

on hold. Therefore, it is expected that, as in the case of Northrop Grumman, current production is below current and anticipated capacity.

Lockheed Martin (Morristown, New Jersey) is a less significant producer of T/R modules, but they anticipate growth. They have produced over 33,000 modules on the COBRA (Counter-Battery). They also have a Navy contract for producing an S-band prototype phased array, which will require over 5,000 modules. However, few have been produced to this point.

Boeing (Seattle, Washington) has entered the T/R module market and is making claims of being able to produce modules at a cost of about $800. However, IDA has neither visited the Boeing facility nor obtained Boeing technical or cost data.

On the demand side, various programs are on the immediate horizon but are not yet creating substantial demands, such as the F-22, JSF, F-18, Cobra Judy, and MDA programs. F-18 radar will go through Milestone C in June, but only for the low-rate initial production quantities. The other programs will not require significant quantities for some time. Therefore, the producer capacities described above should, in the near term, be adequate to avoid bottlenecks.

References

[1] Bruce R. Harmon and Neang I. Om, "Assessing Acquisition Schedules for Unmanned Spacecraft," Institute for Defense Analyses, Paper P-2766, April 1993.

[2] Barry Boehm, *Software Engineering Economics*, Prentice-Hall, Inc.: Englewood Cliffs, New Jersey, 1981.

[3] Space and Missile Systems Center Software Database, *Data Collection Dictionary*, p. C-9.

[4] Jet Propulsion Laboratory, California Institute of Technology, "NASA Flight Project Software Database Data Books," Vols. 1 and 2, January 1989.

[5] A. S. Golberger, *Topics in Regression Analysis*, The Macmillan Company: London, 1968, pp. 120–123.

[6] S. D. Conte, H. E. Dunsmore, and V. Y. Shen. *Software Engineering Metrics and Models*, Benjamin/Cummings Publishing Company, Inc.: Menlo Park, California, 1986, p. 258.

Abbreviations

AMEC	Advanced Microwave Electronics Center
C^3	command, control and communications
CSCI	Computer Software Configuration Item
CSC	Computer Software Component
CSU	Computer Software Unit
DoD	Department of Defense
EMD	Engineering, Manufacturing, and Development
ESLOC	equivalent source lines of code
ESSS	Electronic Sensors and Systems Sector
FQT	First Qualification Test
HPD-X	High Powered Discriminator-X
IDA	Institute for Defense Analyses
JSF	Joint Strike Fighter
KSLOC	thousand source lines of code
MDA	Missile Defense Agency
MIRFS/MFA	Multifunction Integrated RF System/Multifunction Nose Array
MMI	man-machine interface
OLS	ordinary least squares
OTE	Operational Test and Evaluation
SAR	Special Access Required
SLOC	source lines of code
SMC	Space and Missile Systems Center
SSCAG	Space Systems Cost Analysis Group
SRR	System Requirements Review
TER	time-estimating relationship

THAAD	Theater High Altitude Area Defense
T/R	transmit/receive